

Global Canopy Atlas pipeline

Fabian Fischer et al.

2025-04-29

- 1 Package description
- 2 General workflow diagram
- 3 Input data
- 4 Parameters
- 5 Output products
- 6 Future work

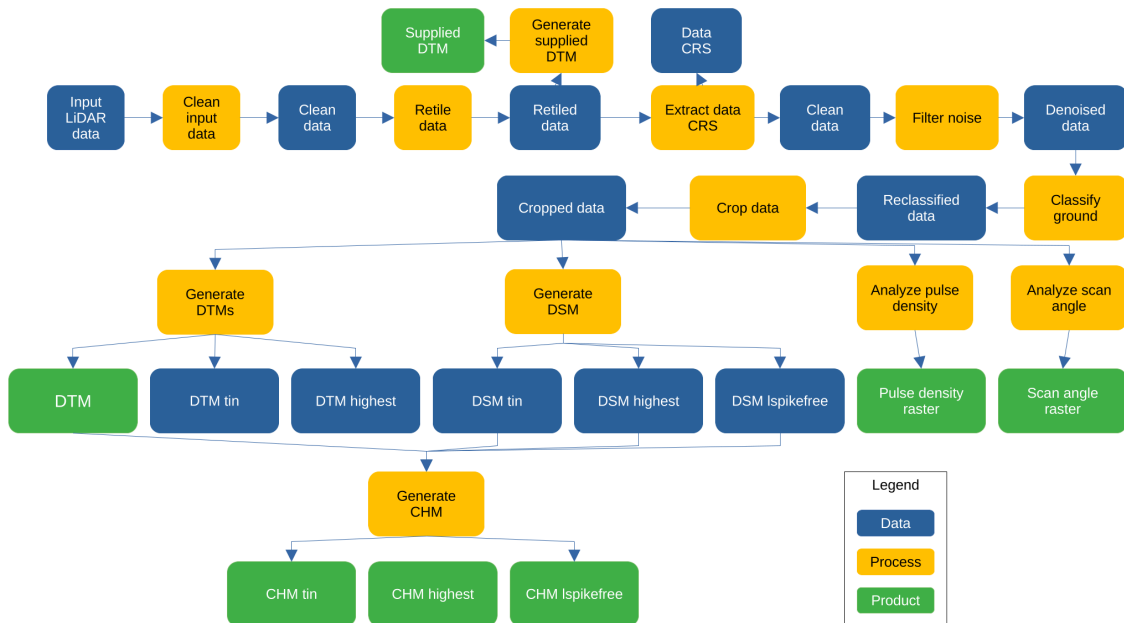
1 Package description

The GCA pipeline is a tool used for the generation of Digital Terrain Model (DTM) and Canopy Height Model (CHM) from Aerial LiDAR Scanning (ALS) data.

The package source code can be found at: https://github.com/fischer-fjd/GCA/tree/GCA_open_source

2 General workflow diagram

The following figure shows a diagram of the pipeline workflow.



In this figure, the blue items represent data being processed by each step, the orange items represent the processing steps of the pipeline and the green items represent the final output products of the pipeline.

3 Input data

The input data is provided in the form of a folder containing a set of pointcloud files, in format LAS (.las) or its compressed equivalent LAZ (.laz). Normally each pointcloud file corresponds to a tile of the scanned landscape. All tiles must have the same Coordinate Reference System (CRS), which should be UTM compatible.

A metadata file should be provided with the data in the same folder. This file shall contain:

- The data CRS
- etc

4 Parameters

- name_job: overall job name, used for processing stats
- type_file: type of the files to be processed, needs to be exact (las, laz, LAS, etc.)
- dir_dataset: folder that contains data sets
- dir_processed: folder where processed data sets should be

saved

- path_lastools: folder to most recent lastools installation
- tmpdir_processing: folder where processing occurs: files will be overwritten and should never be a folder that is synchronized or has slow read/write operations, i.e., no Dropbox folders, no OneDrive, and not an external hard drive resolution = 1.0 resolution of raster products (in m)
- n_cores: number of cores for processing, keep 1-2 cores available for system operations
- size_tile: retiling size
- size_buffer: 25m - 50m, 50 m should be sufficient for any type of acquisition (25m may be too small for ground point classification in sparse scans)
- force.utm: force reprojection of system into UTM (and meter) coordinates; necessary for all files that are registered in feet, otherwise output will be in feet
- force.recompute: force reprocessing; usually set to FALSE, useful when computation has been interrupted for external reasons (power cutoff) and needs to be restarted, because only unprocessed data subsets will be reprocessed
- remove.vlr: probably not necessary in most cases, but should be generally activated. The option leads to the removal of all vlrs AFTER the CRS of the first raster product is set. The CRS will be transmitted to all other raster products, so outputs will not be affected. By deactivating vlrs afterwards we prevent problems with further terra or lastools processing due to odd projection information, which sometimes causes an excess of warnings/errors (and shutdown of parallel processing) or blast2dem to fail
- remove.evlr: probably not necessary in most cases, but should be generally activated. The option leads to the removal of all evlrs AFTER the CRS of the first raster product is set. The CRS will be transmitted to all other raster products, so outputs will not be affected. By deactivating evlrs afterwards we prevent problems with further terra or lastools processing due to odd projection information, which sometimes causes an excess of warnings/errors (and shutdown of parallel processing) or blast2dem to fail
- use.blast2dem: should be activated by default as it improves (or

makes possible) TIN construction in very dense point clouds,
but: not tested on Linux so far

- `type_architecture`: only needed if 32bit Windows should be forced; 32 can be a bit more permissive, particularly for `blast2dem`

5 Output products

- Supplied DTM
- DTM
- CHM
 - TIN
 - Highest
 - Lspikefree
- Pulse density raster
- Scan angle raster

6 Future work

1. Complete integration of the new ground classification algorithm in the pipeline

The new ground classification algorithm is intended to become a more performant open source option compared to the existing algorithms in `lidR`/`lasR`.

This new algorithm has been tested in a standalone approach, and validated with real data. The next step is to integrate it in the pipeline to have a fully open source pipeline with performances closer to the commercial software equivalents (`lastools`).

To do this, two solutions are possible

- Direct call to the pre-compiled executable from R, as done with `lastools` functions
- Wrap the C++ code in R to be able to be used directly from the R pipeline

2. Re-factorization of the pipeline into subsets of modular

functions

The current state of the pipeline has all the sub-functions coded in a common source file. This lead to a complicated readability and maintenance of the code base. A restructuring of this code into individual source files for each function will be done to better organize the repository and have a faster maintenance of the pipeline.

3. Integration and test of the pipeline in the MAAP platform

The Multi-Mission Algorithm and Analysis Platform is a data storage and computing platform put into service by ESA. This platform is available for the FRM4BIOMASS/GEO-TREES actors. The platform will be used to deploy the pipeline and eventually store the data from the project. The pipeline is, to our current knowledge, not still operative for production use. Once this is the case, a deployment and test of the pipeline on the MAAP platform will be done. This will allow us to have a unified reference point for the storage and access to the data, and to efficiently process the outputs of the pipeline.

4. Consolidation of the pipeline output products

The pipeline, in its actual form, produces several versions of DTM and CMH products. This has been done in order to compare different methods, as each version produces the product with a different algorithm. As the pipeline becomes mature enough, a unique algorithm will be selected for each product. The other products will be optionally produced, but not by default.

5. Formal comparison and potential merge with Maryland's university pipeline

In the last stages of the development, a collaboration with the university of Maryland has been established in the context of the GEO-TREES project. The team from Maryland works on their own pipeline, open-sourced, but with different approaches compared to the ones used in our pipeline (notably for the DTM and CHM algorithms).

The pipeline source code and documentation can be found at:

https://github.com/GEO-TREES/ALS_Panama

6. Integration of the pipeline with the AGB estimation

- Production ready – Landscape statistical up-scaling (BIOMASS)

Use the BIOMASS package to, using plot data of the same region scanned by the ALS data, calibrate the allometric statistical model use dfor the upscaling from the plot data to the landscape data provided by the CHM.

More information on this approach and the package itself can be found on: <https://umr-amap.github.io/BIOMASS/index.html>

- Experimental – Individual tree based simulation (CanopyConstructor)

Use the CanopyConstructor package, which uses the CHM and plot data to find correspondences between simulated trees and the landscape data, forming a simulated forest that can be used to generate AGB plots at any desired raster resolution.

More information on this approach and the package itself can be found on: <https://github.com/fischer-fjd/CanopyConstructor>